普通高等教育计算机规划教材

C/C++程序设计教程

刘振安　编著

提供电子教案

机械工业出版社
CHINA MACHINE PRESS

普通高等教育计算机规划教材

C/C++程序设计教程

刘振安　编著

机械工业出版社

本书主要以C/C++语言介绍面向过程语言的编程特点及基本算法。书中介绍了常用的逻辑求解、迭代、递推和递归等问题，以便培养解决实际问题的能力。每章还配备相应的实验和习题，并通过典型例题分析降低学习难度。

全书共12章。第1章C/C++语言的面向过程程序设计；第2章C++语言的基础知识；第3章结构化编程基础；第4章计算机解题和程序调试；第5章构造类型初探；第6章函数与多文件编程；第7章函数、函数指针和多维数组；第8章常用算法实例；第9章结构和链表；第10章使用对象和函数模板；第11章流类库和文件；第12章课程设计实例。

本书适当引入了C++的新特点、STL库和对象的概念，既简化了过程设计，又使读者掌握了基于对象的编程方法。因为不涉及如何设计类，所以既适合作为程序设计的第一门课程，也为将来学习如何设计类创造有利条件。本书特别适合作为高等院校的教材，也可以作为培训班教材，自学教材及工程技术人员的参考书。

图书在版编目（CIP）数据

C/C++程序设计教程 / 刘振安编著. —北京：机械工业出版社，2008.5
（普通高等教育计算机规划教材）
ISBN 978-7-111-24397-7

Ⅰ. C… Ⅱ. 刘… Ⅲ. C语言—程序设计—高等学校—教材 Ⅳ. TP312

中国版本图书馆CIP数据核字（2008）第090884号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）
责任编辑：张宝珠
责任印制：李 妍

北京富生印刷厂印刷

2008年8月第1版·第1次印刷
184mm×260mm·19印张·470千字
0001—5000册
标准书号：ISBN 978-7-111-24397-7
定价：31.00元

凡购本书，如有缺页，倒页，脱页，由本社发行部调换
销售服务热线电话：（010）68326294
购书热线电话：（010）88379639 88379641 88379643
编辑热线电话：（010）88379753 88379739

出 版 说 明

信息技术是当今世界发展最快、渗透性最强、应用最广的关键技术，是推动经济增长和知识传播的重要引擎。在我国，随着国家信息化发展战略的贯彻实施，信息化建设已进入了全方位、多层次推进应用的新阶段。现在，掌握计算机技术已成为21世纪人才应具备的基础素质之一。

为了进一步推动计算机技术的发展，满足计算机学科教育的需求，机械工业出版社聘请了全国多所高等院校的一线教师，进行了充分的调研和讨论，针对计算机相关课程的特点，总结教学中的实践经验，组织出版了这套“普通高等教育计算机规划教材”。

本套教材具有以下特点：

（1）反映计算机技术领域的新发展和新应用。

（2）注重立体化教材的建设，多数教材配有电子教案、习题与上机指导或多媒体光盘等。

（3）针对多数学生的学习特点，采用通俗易懂的方法讲解知识，逻辑性强、层次分明、叙述准确而精炼、图文并茂，使学生可以快速掌握，学以致用。

（4）符合高等院校各专业人才的培养目标及课程体系的设置，注重培养学生的应用能力，强调知识、能力与素质的综合训练。

（5）适合各类高等院校、高等职业学校及相关院校的教学，也可作为各类培训班和自学用书。

机械工业出版社

前　言

本书是在自1994年以来开设的几门课程的基础上，进行合理组合与取舍，并力求反映学科发展，展现它们的最新特征。目前大多数语言教材都是把重点放在基本词法、语法和简单的程序上，学完课程之后，很难编出实用的程序。本书把重点放在程序设计方法上，为了方便学习，每一章均配有相应的实验和习题。在编写中力求取材新颖、结构合理、概念清楚、语言简洁、通俗易懂、实用性强，易于教学。特别适合作为高等院校的教材，也可以作为培训班教材，自学教材及工程技术人员的参考书。

本书讲授面向过程程序设计方法，以C/C++语言介绍面向过程语言的编程特点。因C++兼容C，所以就不再使用C/C++标识，而简称为C++过程程序或C++程序，甚至有时只提算法而不涉及具体的语言实现。书中还专门介绍常用的逻辑求解问题、排序、查找、蒙特卡罗法、枚举、迭代、递推和递归等问题，以便培养解决实际问题的能力。因为面向对象编程也离不开函数及其算法，所以加强算法研究，能为学习面向对象程序设计打下良好基础。尤其是第12章的课程设计，更好地演示了多个函数和多文件编程的奥秘，更贴近面向对象的思想，为学习面向对象和可视化的多文件编程模式奠定基础。每章还配备相应的实验和习题，并通过典型例题和错误分析以降低学习难度。因此，本书也是为突破目前教学格式的尝试。

本书共分12章。第1章是C/C++语言的面向过程程序设计，主要是引入构成C/C++语言程序设计的教学模式。第2章是C++语言的基础知识，重点引入C++语言基本数据类型和表达式，为学习面向过程编程打下基础。第3章是结构化编程基础，将结合实例介绍结构化程序的基本设计原理。第4章是计算机解题和程序调试，主要结合实例，说明使用计算机解决具体问题的思路以及程序调试方法。第5章是构造类型初探，将探讨几个典型的构造类型，并简单说明它们的使用方法。第6章是函数与多文件编程，重点讨论多个文件中的函数调用问题。第7章是函数、函数指针和多维数组，本章要用到第5章和第6章的知识，并将它们有机地结合起来。第8章是常用算法实例，将结合典型算法(逻辑求解、迭代、递推和递归等)简要介绍基于过程编程的基本方法。注意有些例题将使用数组知识，为了练习数组的使用，还将继续给出求解逻辑问题的例子。第9章是结构和链表，本节的重点是介绍结构类型及其质的变化。第10章是使用对象和函数模板，本章将介绍如何使用对象和模板编制面向过程的程序，以提高编程效率。第11章是流类库和文件，有了第9章和第10章有关类及其成员函数的知识，就可以直接使用这些知识学习本章，从而达到事半功倍的效果。第12章是课程设计实例，本章课程设计是使用数组设计一个实用的小型学生成绩管理程序。它具有查询和检索等功能，并且能够对指定文件操作，也可将多个文件组成一个文件。本课程设计涉及内容较多，也可以根据实际教学情况决定。附录提供C语言的输入输出格式。

由于本人才疏学浅，不妥之处再所难免，敬请各位不吝赐教，给予指正为盼。

刘振安

目　录

第1章　C/C++语言的面向过程程序设计

本章首先使用伪码，以设计一个输入三角形的3个顶点坐标、计算3条边的长度的算法为例，介绍基于过程的程序设计的基本概念。然后简要介绍C/C++语言的特点，通过简单而典型的C/C++语言实例，引入本书构成C/C++语言程序设计的教学模式，从而建立使用C++语言设计面向过程程序的基本概念。

1.1　面向过程与结构化程序设计

本节将简要介绍C语言的发展过程及其特点，面向过程的编程特点及结构化程序设计的基础知识。

1．C语言的特点

C语言是20世纪70年代初期美国贝尔（Bell）实验室 Dennis M.Ritchie 设计的一种程序设计语言，正式发表于1978年。

1970年，Ken Thompson 在早期编程语言 BCPL 的基础上开发了一种新的语言，取名叫B。Dennis M.Ritchie 在B的基础上，于1971年开发了第一个C编译程序，1972年开始使用（主要是在贝尔实验室内部使用）。以后，C 语言又经过多次改进，直到1975年用C语言编写的 UNIX 操作系统第6版公诸于世后，C 语言才举世瞩目。目前，其应用领域已不再限于系统软件的开发，而成为当前最流行的程序设计语言之一。

1978 年，Brian Kernighan 和 Dennis M.Ritchie 在 C 程序语言（The C Programming Language）一书中对C语言作了详尽的描述。随着微型计算机的日益普及，大量的C语言工具相继问世，然而这些工具没有统一的标准，并有不一致的现象。为了改变这种情况，ANSI于1983年成立了一个专门委员会，为C语言制定了ANSI标准。当时比较流行的有TURBO C，它不仅满足 ANSI 标准，还提供了一个集成开发环境，同时也按传统方式提供了命令行编译程序版本以满足不同用户的需要。随着 Windows 编程的兴起，Borland C 和 Microsoft C 受到用户的欢迎。目前比较流行的是兼容C语言的 Microsoft Visual C++ 6.0 及 BorLand C++ 集成环境。

C 语言是一种通用的程序设计语言。C 语言的通用性和无限制性，使得它对许多程序设计者来说都显得更加通俗，更加有效。目前C语言已用于各个方面的程序设计，无论设计系统软件（操作系统，编译系统等）或应用软件（图形处理），数据处理（如企业管理）或数值计算等都可以很方便地使用C语言。C语言有如下特点：

1）C语言吸取了汇编语言的精华，使C语言对高级语言来讲是“低级”语言（汇编语言是一种面向机器的程序设计语言，尽管它的编程相对高级语言来要麻烦得多，但由于它具有描述准确和目标程序质量高的优点，所以汇编语言仍然有很强的生命力）。

① C语言提供了对位、字节以及地址的操作，使程序可以直接对内存及指定寄存器进行操作。

② C 语言吸取了宏汇编技术中的某些灵活的处理方法，提供宏代换 #define 和文件蕴含 #include 的预处理命令。目前 ANSI 新标准采用了 C++的 const 类型限定符，更增加了可靠性。

③ C 语言能很方便地与汇编语言连接。在 C 程序中引用汇编程序与引用 C 语言函数一样，这为某些特殊功能程序的设计提供了方便。

2）C 语言继承和发扬了高级语言的长处，使 C 语言相对汇编来讲又是“高级”语言。

① C 语言吸取了 ALGOL 的分程序结构思想。C 程序中，可用一对花括号 “{}”把一串语句括起来而成为复合句（分程序），在括号内可定义变量。它还继承了 PASCAL 的数据类型，提供了相当完备的数据结构。

② C 语言吸取了 FORTRAN 语言的模块结构思想。C 程序中，它的每一个函数都是独立的，可以单独编译。对设计一个大的程序来说，有利于分工编程和调试。

③ C 程序中的任何函数都允许递归，这样对某些算法实现起来就十分方便。

3）C 语言的规模适中、语言简洁，其编译程序简单、紧凑。C 语言在表示上尽可能地简洁（比如用一对花括号 {} 代替 Begin End，运算符尽量缩写等），C 语言本身没有提供输入和输出工具以及并行操作，它的许多成分都是通过函数调用来完成的，而且运行时所需要的支持少，占用的存储空间也小。

4）C 语言的可移植性好，这是指程序从一个环境不加或稍加改动就可以搬到另一个完全不同的环境下去运行。汇编程序因依赖机器硬件，所以根本不可移植；而一些高级语言，如 FORTRAN 等编译的程序也是不可移植的。

5）生成的代码质量高，在代码效率方面可以和汇编语言相媲美。

C 语言的优点很多，但也有些不足之处。例如：运算符优先级太多，不便记忆；有些还与常规约定有所不同；类型检验较弱，转换比较随便，不太安全等。尽管如此，由于上述几个突出的优点，C 语言仍不失为一个实用的通用程序设计语言，因此学习和使用它的人越来越多。

2. 典型的面向过程编程语言

C 语言是典型的面向过程的编程语言。所谓“面向过程”，就是不必了解计算机的内部逻辑，而把精力主要集中在对如何求解问题的算法逻辑和过程的描述上，通过编写程序把解决问题的步骤“告诉”计算机。

【例 1-1】 给出输入三角形的 3 个顶点坐标，计算 3 条边的长度的过程算法描述。

从面向过程的角度看，问题的实质是取得 3 个顶点的坐标，然后计算每两点之间的距离。可以将求解过程简单地描述如下。

输入：3 个坐标，即 6 个数据。

输出：3 条边的长度。

算法设计：

接受 3 组数据，每组 2 个数据。

(x1,y1) ← 第 1 组数据

(x2,y2) ← 第 2 组数据

(x3,y3) ← 第 3 组数据

计算每两点之间的距离

AB ← (x1,y1) 与 (x2,y2)之间的距离

AC ← (x1,y1) 与 (x3,y3)之间的距离
BC ← (x2,y2) 与 (x3,y3)之间的距离
输出（AB，AC，BC）

由此可见，这种算法的特点是按部就班地求解,而且条理清晰易懂。

随着应用需求的扩大和变化，软件生产的方法和效率仍然远远跟不上社会发展的需要。软件工作者开始考虑如何提高软件质量和生产效率，即使用系统方法代替个人经验、智慧、技巧等，使建立软件系统的过程遵从一系列规范化阶段，包括需求分析、概要设计、详细设计、实现、组装测试、运行和维护等。这就将软件设计工作推进到软件工程时代。

结构化程序设计被称为软件发展中的第 3 个里程碑，其影响将比前两个里程碑（子程序、高级语言）更为深远。结构化程序设计的概念，最早是由 Dijkstra 提出的。他在 1965 年召开的 IEIP 会议上提出“GOTO 语句可以从高级语言中取消”，“一个程序的质量与程序中所含的 GOTO 语句的数量成反比”。

Tom.De Marcod 在《结构化分析与系统规格说明》教材中提出基于模型的软件工程概念，认为对于复杂软件系统的创建，必须首先为它们建立系统的书面模型。另一个有影响的软件理论是 Niklans Wirth 提出的“算法+数据结构=程序”。将软件划分成若干个可以单独命名并分别编写的部分，称其为模块。模块化使软件能够有效地管理、维护、分析和处理复杂问题。在 20 世纪 80 年代，软件工作者普遍接受了模块化的程序设计方法，接下来就是争论如何建立模块。有人认为最佳途径是使用函数，有人认为每个模块只应容纳一个数据结构，有人认为每个模块只应做一件事，还有人提出了事件驱动概念。这些代表性解决方案也促进了程序设计语言的发展。

C 语言是结构化程序设计语言，它的程序设计特点就是函数设计。所谓函数，就是模块的基本单位，是对处理问题的一种抽象。例如，将求绝对值的功能抽象为 abs（参数），就有 abs(5)=5 和 abs(−55)=55，称 abs 为求一个数的绝对值函数，而称 5 和−55 为函数 abs 的参数。把一切逻辑功能完全独立的或相对独立的程序部分都设计成函数，并让每一个函数只完成一个功能。这样，一个函数就是一个程序模块，程序的各个部分除了必要的信息交流之外，互不影响。相互隔离的程序设计方法就是模块化的程序设计方法。C 语言的这种程序结构化和模块化设计方法，特别适合于大型程序的开发。它解决了过去组成大系统时所产生的多文件的组织与管理问题。

在程序规模比较大时，一般是根据结构化程序设计方法将程序划分成多个源文件。在编译该程序时，可以按一个个源文件为单位，分别进行编译并产生与之对应的目标文件，然后再用连接程序把所生成的多个目标文件连接成一个可执行文件。称 C 语言的这种编译过程为分块编译。

C 语言的这种分块编译处理方式可以使一个程序同时由多个人进行开发，这为大型软件的集体开发提供了有力的支持。分块编译的优点还在于修改一个源文件中的程序后，并不需要把整个程序的所有文件重新编译，这就大大节省了时间。

3. C++兼容面向过程编程

下面是用 C++语言编写一个求三角形两点之间距离、面向过程的算法思想。

【例 1-2】 给出输入三角形的 3 个顶点坐标，使用 C++语言编写计算 3 条边的长度的过程算法描述。

输入：3 个坐标，即 6 个数据。

输出：3 条边的长度。

算法设计：

接受 3 组数据，每组 2 个数据

(x1,y1) ← 第 1 组数据

(x2,y2) ← 第 2 组数据

(x3,y3) ← 第 3 组数据

计算每两点之间的距离

AB ← func(x1,y1,x2,y2)

AC ← func(x1,y1,x3,y3)

BC ← func(x2,y2,x3,y3)

输出（AB，AC，BC）

// 函数 func 的说明

func(a,b,c,d)

参数：a,b,c,d

功能：求$(a-c)^2+(b-d)^2$的平方根

函数 func 相当于 C++函数库中的 sqrt 函数，可以直接使用 sqrt 函数求平方根。例如：

AB ← sqrt((x1−x2)*(x1−x2)+ (y1−y2)*(y1−y2))

由此可以看出，面向过程的程序设计的关键是考虑使用结构化设计方法，使程序模块化。因为它是以函数过程和数据结构为中心，所以不能直接反映出人类认识问题的过程。

1.2 面向对象与面向过程

1. 面向对象编程的基本概念

程序设计语言是计算机科学中一个不断变化的领域。有的语言已退出舞台，让位给新的语言(如 Smalltalk, ML, Prolog)。有的语言，如 FORTRAN，则不断地演化发展着，以巩固自己的地位。同时，人们又力求开发出更好的程序设计语言。

结构化程序设计技术是 20 世纪 70 年代研究的中心问题，而今天的热点则是面向对象程序设计语言，如 Smalltalk，Traits，Eiffel 和 Java 等语言都是面向对象的程序设计语言。C++则是在标准 C 语言的基础上，引入“面向对象”概念而扩充形成的混合型面向对象语言。

面向对象的程序设计方法不是以函数过程和数据结构为中心，而是以对象代表求解问题的中心环节。它追求的是现实问题空间与软件系统解空间的近似和直接模拟。这就改变了原来计算机程序的分析、设计和实现的过程与方法之间的脱节和跳跃状态，从而使人们对复杂系统的认识过程与系统的程序设计实现过程尽可能地一致。

面向对象方法的产生，是计算机科学发展的要求。20 世纪 80 年代，特别是 90 年代以来，软件的规模进一步扩大，对软件可靠性和代码可重用性的要求也进一步提高，就是在这样的背景下，面向对象的程序设计方法应运而生。

【例 1-3】 给出输入三角形的 3 个顶点坐标，计算 3 条边的长度的面向对象的算法描述。

根据“问题域”设计程序模块。这里首先从顶点坐标考虑，顶点坐标就是一个点，通过

这个点，可以得到它与另一个点的距离，从而将平面抽象为点对象的集合。三角形的 3 个顶点就是点类的 3 个实际对象。求三角形的 3 条边长，就是求每两个点对象之间的距离。因此，重点应放在如何描述这个点类上。

设计 point 类，这个类具有两个坐标值（x,y），称为类的属性。为类设计一个与类同名的特殊函数 point，point 表现了类的特定行为，即用来初始化这个点的属性值，能使不同点的对象具有不同的 x 和 y 值（也就是不同的属性值）。这个点类能向外界提供自己的属性值，并能计算它与另一个对象之间的距离。由此推知，可以像图 1-1 那样描述这个类。

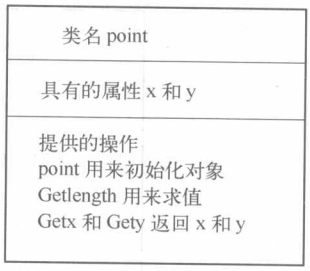

图 1-1　point 类示意图

第 1 个方框中是类名，第 2 个方框中是坐标点的数据，称为属性（或称数据成员）。第 3 个方框中表示类所提供的具体操作方法，实际上是如何使用数据 x 和 y，以实现预定功能的函数，这里称为成员函数。point 是一种特殊的成员函数，用来初始化类的数据成员的值，从而产生一个具体的对象。如果需要定义对象 A，使用的方法如下：

```
point A(x1,y1);
```

这里 x1 和 y1 是对象 A 的坐标。对于 A 和 B 之间的距离，既可以表示为 A.Getlength(B)，也可表示为 B.Getlength(A)。这个道理很明显，只要表示为两个点的对象即可。至于是 A 发出求 A 与 B 之间的距离的消息，还是 B 发出求 B 与 A 之间的距离的消息，则是无关紧要的。它们是调用同一个消息处理函数 GetLength。算法描述为：

输入：3 个类的对象。

输出：任意两个对象之间的距离。

算法设计：

```
设计类 point
    类的结构图如图 1-1 所示
产生 3 个对象
    point A(x1,y1) ←对象 A
    point B(x2,y2) ←对象 B
    point C(x3,y3) ←对象 C
计算对象之间的距离
    AB ← A.Getlength(B)
    AC ← A.Getlength(C)
    BC ← B.Getlength(C)
输出（AB，AC，BC）
```

求解的中心环节是以点这个客观世界的对象为依据，这正符合人们对客观世界的认识。

需要强调的是，面向对象语言和面向对象编程不能简单地等同起来。正像不能说采用结构化程序设计语言编写的程序一定符合结构化编程的特点一样，并不能说采用面向对象语言的编程就一定具有充分的面向对象特性。同样，采用面向过程的语言，仍然可以实现面向对象编程。最早的 Windows 编程，就是使用 C 语言实现的。图 1-2 所示是使用 C 语言编制的典

型 Windows 程序界面。可以在图 1-2 所示的界面上完成输入并得到输出结果，并且可以反复输入数据进行计算，如果有兴趣，还可以同时将这个三角形的图形绘出来。

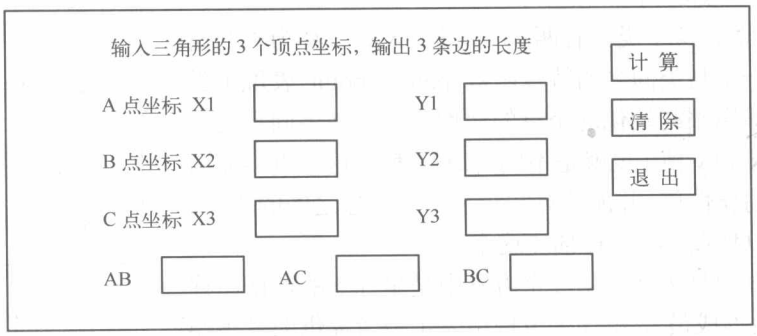

图 1-2　用 C 语言编制的典型 Windows 程序界面

时至今日，仍然需要使用 C 语言编制 Windows 程序，尤其是设备驱动等程序。

结构化程序设计的提出和发展主要是为了满足日益复杂和规模庞大的软件开发的要求。因为 C++是混合型语言，所以可以使用 C++编译器所提供的对象设计出更好的、面向过程的软件系统。

随着软件的进一步庞大和复杂，随着编程实践的深入，人们发现，当软件复杂到一定程度时，结构化程序设计也不足以满足需要。一般说来，当软件的规模在三四万行以内时，结构化的方法还是可以满足需要的，当程序的规模超过这个尺度时，开发和维护就会变得越来越困难。发生这种情况的根本原因是结构化程序设计方法与客观世界以及人们的分析思考方式都非常不一致。这种不一致实际上是不合理的一种表现。这种不合理性的一个具体结果是，结构化程序设计中的分而治之的想法尽管非常好，但在结构化程序设计语言和结构化程序设计方法下却难以贯彻到底。比如，结构化程序设计要求尽量不用全局变量，但当程序规模大到一定程度时，以功能抽象为基础的结构化程序几乎不可避免地引入大量的全局变量。也就是说，实际上已经没有办法满足结构化程序设计的基本要求。而在面向对象程序设计中，可以将一组密切相关的函数统一封装在一个对象中，从而可以合理有效地避免全局变量的使用。可以认为，面向对象方法更彻底地实现了结构化程序设计的思想。

结构化程序设计使用的是功能抽象，面向对象程序设计不仅能进行功能抽象，而且能进行数据抽象。“对象”实际上是功能抽象和数据抽象的统一。

2．自然语言与计算机语言之间的鸿沟

软件开发是对给定问题求解的过程。从认识论的角度看，可以归为两项主要活动：认识与描述。软件开发者将被开发的整个业务范围称作“问题域”(Problem Domain)，“认识”就是在所要处理的问题域范围内，通过人的思维，对该问题域客观存在的事物以及对所要解决的问题产生正确的认识和理解，包括弄清事物的属性、行为及其彼此之间的关系并找出解决问题的方法。因为人类的任何思维活动都是借助于他们所熟悉的某种自然语言进行的，所以开发人员对问题域的认识是人类的一种思维活动。我们常常看到有人讲话很流利，但考虑问题时却很糊涂，这说明还需要具有正确的思维方法。尤其是在软件开发过程中，要求人们对问题域的理解，要比日常生活中对它的理解更深刻、更准确。这需要许多以软件专业知识为